PAL powers universal ISA bus interface

JERZY R CHRZĄSZCZ, INSTITUTE OF COMPUTER SCIENCE, WARSAW UNIVERSITY OF TECHNOLOGY, NOWOWIEJSKA, WARSAW, POLAND

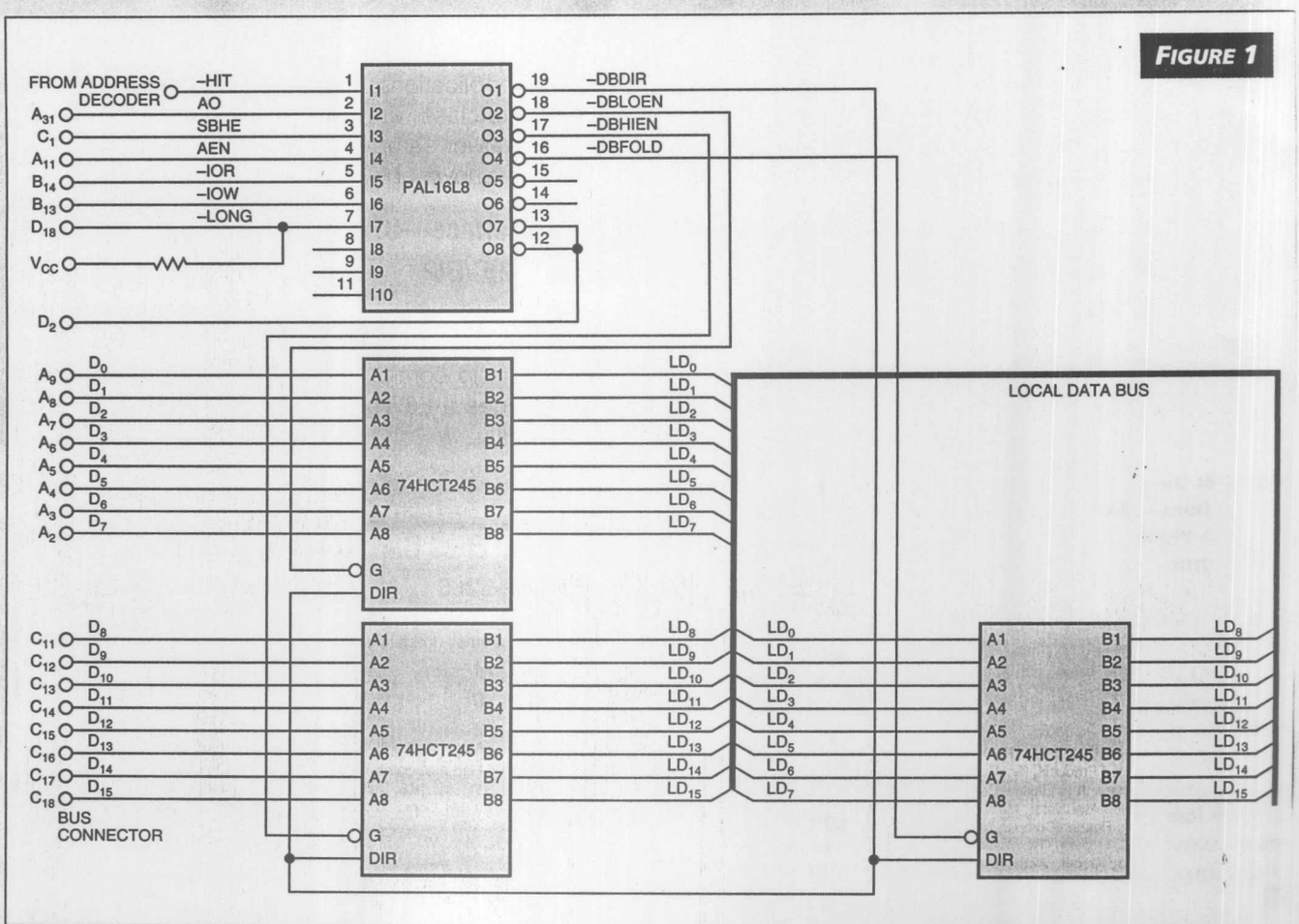
A pc board bearing the 16-bit ISA databus interface in **Fig 1** can adapt automatically to either 8- or 16-bit motherboard slots. The interface comprises three bidirectional octal buffers and glue logic. The glue logic controls transfer direction and output enables.

The logic integrated into the PAL16L8 distinguishes between 8- and 16-bit cycles using the state of the following signals: the LSB address bit (AO), the bus-high-enable flag (SBHE), the address-enable (AEN), and I/O-access strobes (–IOR and –IOW). A single resistor across V_{CC} and pin D_{18} of the ISA's secondary socket senses if the user has

LISTING 1—GLUE-LOGIC PAL PROGRAM

```
dbdir =>
    ior

dbfold =>
    a0 & !aen & hit & !ior & iow & !long
  # a0 & !aen & hit & ior & !iow & !long

dbhien =>
    !aen & hit & !ior & iow & long & !sbhe
  # !aen & hit & ior & !iow & long & !sbhe

dbloen =>
    !aen & hit & ior & !iow
  # !aen & hit & !ior & iow

iord16 =>
    1

iord16.oe  =>
    !a0 & !aen & hit & ior & !iow & long & !sbhe

iowr16 =>
    1

iowr16.oe  =>
    !a0 & !aen & hit & !ior & iow & long & !sbhe

read =>
    !aen & hit & ior & !iow

word =>
    !a0 & long & !sbhe

write =>
    !aen & hit & !ior & iow

dbdir.oe  =>
    1

dbfold.oe  =>
    1

dbhien.oe  =>
    1

dbloen.oe  =>
    1
```

FIGURE 1



This simple design forms the core of a universal ISA bus interface.

plugged the board into an 8- or 16-bit slot.

The PAL generates the signal –IOCS16 to indicate the ability to make 16-bit transfers. If, during a 16-bit I/O access, the board does not assert –IOCS16, the motherboard's bus controller performs an additional byte cycle, allowing the high data byte to pass via bus lines D_0 through D_7.

This design is only the core of the interface (note the several unused PAL pins). But, starting with it, you can develop more sophisticated functions, assert wait states, support DMA cycles, and so on. If you migrate the design to 24-pin PLDs, you could even afford an on-chip, 10-bit I/O-address decoder. (DI #1648) EDN

To Vote For This Design, Circle No. 347

Program flags dangerous DOS commands

M N JAYARAM, BARC, BOMBAY, INDIA

BBS Windows users still perform many small housekeeping tasks, such as renaming, moving, or deleting files, by switching to DOS, because DOS is much faster than the time-consuming, icon-driven facilities of Windows.

However, you should not use certain DOS features, such as *CHKDSK* or tape backup, when Windows is running. Windows, because it is a multitasking operating system, maintains several open and swap files. Any changes to the FAT (file-allocation table) or directory structure may result in loss or corruption of data.

You can pre-empt the dangerous DOS commands by providing an appropriate *.COM* driver program to kick off the corresponding *.EXE* file. **Listing 1** shows a debug script, *safedos.scr*. To create the file *CHKDSK.COM*, for example, execute the DOS command:

debug <safedos.scr

Because files with *.COM* extensions receive higher priority than files with *.EXE* extensions, DOS will execute the program *CHKDSK.COM* when you enter *CHKDSK*. To create other *.COM* files to kick off their corresponding *.EXE* files change the lines

db 'c:\dos\chkdsk.exe',0
nchkdsk.com

in debug-script *safedos.scr* appropriately.

An environment variable called *windir* exists when Windows is running. This environment variable is unique because it is the only environment variable stored in lower case. All other environment variables are in upper-case characters. (DOS's *SET* command can create environment variables in upper case only.) The program scans the "environment" (DOS-ese for a reserved section of memory) for the environment-variable *windir*. If found, the program exits with the error message "Shutdown Windows!". Otherwise the program loads and executes c:\dos\chkdsk.exe.

Because a *.COM* program assumes that it is the sole owner of the entire memory, you must free the memory not needed by the program so that it can safely and successfully execute a child process. Before freeing the additional memory you must move the stack pointer to within the program area. The text attached to **EDN BBS /DI_SIG #1650** contains the program and documentation. (DI #1650) EDN

To Vote For This Design, Circle No. 348

LISTING 1—DEBUG-SCRIPT SAFEDOS.SCR

```
a100
mov bx,40
mov sp,3fe
mov wo[3fe],0
mov ah,4a
int 21
mov ax,[2c]
dec ax
mov ds,ax
mov ax,[3]
push cs
pop ds
mov cl,4
shl ax,cl
mov cx,ax
mov bx,2c
mov es,[bx]
mov al,[017b]
xor di,di
repne
scasb
jcxz 0141
es:
mov al,[di]
cmp al,[017c]
jnz 0141
lea dx,[017d]
mov ah,9
int 21
ret
lea dx,[01b2]
push cs
pop es
mov ax,cs
mov [01a8],ax
mov ax,0080
mov [01a6],ax
lea bx,[01a4]
mov [01a0],ss
mov [01a2],sp
mov ax,4b00
int 21
jnc 016d
mov ah,9
lea dx,[0191]
int 21
mov ax,[01a0]
mov bx,[01a2]
cli
mov ss,ax
mov sp,bx
sti
ret
db 'r='
db 7,'Shutdown Windows! $'
db 'Exec failed ! $'
dw 0
dw 0
dw 0
dw 0
dw 0
dw ffff
dw ffff
dw ffff
dw ffff
db 'c:\dos\chkdsk.exe',0

nchkdsk.com
rcx
100
w
```